10-1-2000

MACS: Multi-agent COTR system for Defense Contracting

J. Liebowitz
University of Maryland - Baltimore County

Monica Adya
Marquette University, monica.adya@marquette.edu

B. Rubenstein-Montano
University of Maryland - Baltimore County

Victoria Y. Yoon
University of Maryland - Baltimore County

J. Buchwalter
University of Maryland - Baltimore County

See next page for additional authors

Accepted version. *Journal of Knowledge-Based Systems*, Vol. 13, No. 5 (2000): 241-250. DOI.
NOTICE: this is the author's version of a work that was accepted for publication in *Journal of Knowledge-Based Systems*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Journal of Knowledge-Based Systems*, VOL 13, ISSUE 5, (October 2000). DOI.
Monica Adya was affiliated with the University of Maryland-Baltimore County at the time of publication.

Authors
J. Liebowitz, Monica Adya, B. Rubenstein-Montano, Victoria Y. Yoon, J. Buchwalter, M. Imhoff, S. Baek, and C. Suen

MACS: Multi-Agent COTR System for Defense Contracting

J. Liebowitz[a]

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

M. Adya

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

B. Rubenstein-Montano

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

V. Yoon

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

J. Buchwalter

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

M. Imhoff

Department of Information Systems, Laboratory for Knowledge Management, University of Maryland-Baltimore County, Baltimore, MD

S. Baek

Department of Management and Information Systems, Haub School of Business, St. Joseph's University, Philadelphia, PA,

C. Suen

Center for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Quebec, Canada

Abstract: The field of intelligent multi-agent systems has expanded rapidly in the recent past. Multi-agent architectures and systems are being investigated and continue to develop. To date, little has been accomplished in applying multi-agent systems to the defense acquisition domain. This paper describes the design, development, and related considerations of a multi-agent system in the area of procurement and contracting for the defense acquisition community.

Keywords: Multi-agents; Intelligent agents; Acquisition

1. Introduction

Procurement and contracting are integral parts of the acquisition management process. In US defense research contracting, the Acquisition Request Originator (ARO) and Contracting Officer's Technical Representative (COTR) play important roles in the pre-award and post-award contractual phase. Their responsibilities include evaluating procurement request (PR) packages and identifying forms and other components of the packages that will ensure their completion. These activities require them to be familiar with the policies and procedures that support the acquisition management process. In many U.S. defense laboratories, scientists must participate in the procurement and contracting process in order to be awarded contracts and continue with their work. However, the nature of contracting involves many complex, frequently changing rules and regulations. It is difficult for the ARO/COTR to remember and to keep up-to-date with these new rules/procedures, particularly since he/she is principally a scientist or engineer and not a contract specialist.

These activities often become burdensome and are not part of the actual research effort.

To assist the ARO/COTRs in handling the pre-award phase of a contract, such as putting together a PR package, and many other acquisition concerns/rules/ regulations, the Defense Acquisition Deskbook has been created and appears in both web and CD format (http://www.deskbook.osd.mil). This Deskbook is updated regularly in order to have the most current set of acquisition rules and regulations at the fingertips of the ARO/COTR. The Procurement Desktop-Defense (PD2)/Standard Procurement System (//pd2.amsinc.com) has been also developed as the standard for procurement rules and regulations. A component of the Defense Acquisition Deskbook is the "Ask a Professor" module whereby one submits a question and experts in resource centers reply to these requests. There are typically about 100 questions sent to experts each month. In addition, the Contracting Officer's Technical Representative Expert System Aid (CESA) has been developed to capture the expert's knowledge and experiential learning to help the ARO/COTRs and train new specialists in the pre-award phase of a contract[18].

Although CESA can play valuable roles in assisting in the contracting process, multi-agent technology seems to have potential for enhancing support for ARO/COTRs beyond the capabilities of CESA. Among many features of multi-agent technology, its capabilities for collaboration and adaptation are particularly appealing for this problem domain. First, agents are capable of cooperating and collaborating with other agents and possibly human users to solve problems. Agents share information, knowledge, and tasks among themselves, and cooperate with each other to achieve common goals. The capability of a multi-agent system is not only reflected by the intelligence of individual agents but also by the emergent behavior of the entire agent community[29]. This ability allows each agent to be designed to represent a different specialty area of the Defense Acquisition Deskbook and develop responses to the inquiries on the pre-award phase of a contract through collaboration among multi-agents. Second, agents are capable of adapting to the environment, including other agents and human users. Agents can learn from experience over time to improve their performance[15]. The learning capability is particularly promising for long term use in the contract acquisition area. A multi-agent system can learn appropriate responses based on user inputs and new requirements for contract acquisitions. Such multi-agent technology may be a viable alternative to automate parts of the Ask a Professor component in the Defense Acquisition Deskbook. The multi-agent system called MACS (Multi-Agent COTR System) has been developed to assist in defense acquisition, and is a method for capturing, sharing, and disseminating knowledge as related to the knowledge management field for defense acquisition applications.

Knowledge management[19,20,21] is the process of creating value from an organization's intangible assets. It deals with how best to leverage knowledge internally in the organization and externally to the customers and stakeholders. As such, knowledge management combines various concepts from numerous disciplines, including organizational behavior, human resources management, artificial intelligence (AI), information technology, and the like. The focus is how best to share knowledge to create value-added benefits to the organization.

In looking at ways for sharing knowledge, transforming individual knowledge into collective, organizational knowledge, and reincarnating organizations into "knowledge organizations", the field of AI can help push these basic tenets of knowledge management[30]. One of the important areas of knowledge management is knowledge capture and representation. The knowledge engineering[10] methodologies for building expert systems have applied knowledge acquisition techniques (e.g. interviewing, protocol analysis, simulation, personal construct theory, card sorting, etc.) for eliciting the tacit knowledge from domain experts. In order to develop knowledge repositories in knowledge management systems for formally documenting knowledge in an on-line way, these knowledge acquisition techniques could be applied. Additionally, knowledge discovery and data/text mining approaches (AI-related methods) could be used to inductively determine relationships and trends in these knowledge repositories for creating new knowledge. In order to represent this knowledge in these repositories, a knowledge taxonomy and knowledge mapping are typically constructed for serving as the frameworks on which to build these knowledge repositories. Knowledge ontologies and ways for representing acquired knowledge (rules, cases, scripts, frames/objects, semantic networks, etc.) are typically created in the AI field for building expert and other intelligent systems. The knowledge management field can apply these AI techniques to help codify the knowledge in the knowledge management systems. Other AI techniques like intelligent agents[3] can be used to help in the search and retrieval methods of knowledge in the knowledge management systems. Agents can be used to help in combining knowledge which would ultimately lead to the creation of new knowledge. The AI Applications Institute at the University of Edinburgh has developed an adaptive workflow system, using agent technology, to support knowledge management. Natural language and speech understanding front-ends as interfaces to knowledge management systems may be worthwhile AI techniques to apply in the coming years to the knowledge management field.

Our MACS system uses agent-based technology to enhance the knowledge of those interested in gaining insights into the acquisition field. The objective of this paper is to present the architecture, implementation, and related considerations of a multi-agent system, called MACS. The system is designed to help the ARO/COTR in answering questions about the pre-award phase of a contract. Knowledge for this multi-agent system is extracted from CESA[18]. MACS could ultimately be used in the Ask a Professor module by applying agents to search the Deskbook and develop responses to ARO/COTR related questions.

MACS has been designed using both AgentBuilder® software and a Java servlet. Essentially, the agent that interfaces with users is a Java servlet that can be viewed on the Internet. This agent then communicates with AgentBuilder® where the other agents in the system, and their knowledge from CESA, reside. Communication between the agent designed as a servlet and the AgentBuilder® agents is accomplished with a Java-based communications API provided by Reticular Systems, Inc., the vender for AgentBuilder®. This API makes use of the Remote Method Invocation to access distributed objects over a network.

The next section reviews the literature on multi-agent frameworks. Section 3 presents applications of multi-agent systems in the procurement and contracting/acquisition areas. Section 4 then describes the architecture of MACS for the pre-award phase of contracting and the implementation, and Section 5 summarizes our work.

2. Multi-agent system frameworks

Over the past few years, some interesting work has been developed in creating multi-agent system frameworks[8]. One such framework by DeLoach[6] develops a methodology for multi-agent systems engineering. The framework includes the following[6]:

1. identify agent types;
2. identify the possible interactions between agent types;
3. define coordination protocols for each type of interaction;
4. map actions identified in agent conversations to internal components;
5. define inputs, flows, and outputs associated with the agents;
6. select the agent types that are needed;
7. determine the number of agents required of each type and define: the agents' physical location or address, the types of conversations that agents will be able to hold, and any other parameters defined in the domain.

Zeus, developed at British Telecom Laboratories by Collis et al.,[5] is an advanced toolkit for engineering distributed multi-agent systems. Zeus contains an agent component library, visualization tools, and agent building software. The Zeus agent design methodology is to determine candidate agents, define each agent using the graphical Zeus Generator tool and identify tasks, describe agent relationships using Zeus Generator, choose from a list of prewritten coordination strategies, and implement/encode the agents.

Flores-Mendez,[9] with the Collaborative Agents Group at the University of Calgary, proposes the need for a standardized multi-agent system framework. He describes the multi-agent system as an environment consisting of areas. Areas are required to have exactly one local area coordinator, which is an agent that acts as a facilitator for other agents within its area. Agents use the services of local area coordinators to access other agents in the system. Agents can also be connected with yellow page servers and cooperation domain server agents.[9]

A variety of other work on multi-agent systems has been undertaken. Landauer and Bellman[16] describe an approach to integration in constructing complex systems that rely on cooperative collections of agents instead of a central planner or organizer. Sycara and Zeng[34] discuss the coordination of multiple intelligent software agents. Arisha et al.,[1] from the University of Maryland-College Park, describe a platform called Impact for collaborating agents. Yabrou et al.[14] at the University of Maryland-Baltimore County (UMBC), describe the various agent communications languages — KQML (Knowledge Query Manipulation Language), FIPA ACL (Foundation for Intelligent Physical Agents-Agent Communication Language), and others. Joshi and Singh[12] guest edited a special issue on "Multiagent Systems on the Net" with a myriad of papers looking at multi-agent system frameworks and applications. The HINTS system, developed by Computer Sciences Corporation for the Australian defense/health-care communities is another example of a multi-agent system that has been developed.

Furthermore, Sycara[33] discusses multi-agent systems and the challenges ahead, namely: (1) how to decompose problems and allocate tasks to individual agents; (2) how to coordinate agent control and communications; (3) how to make multiple agents act in a coherent manner; (4) how to make individual agents reason about other agents and the state of coordination; (5) how to reconcile conflicting goals between coordinating agents; and (6) how to engineer practical multi-agent systems. In addition to this list of challenges, many researchers are looking at only autonomous agents; but in many situations, the integration of human collaboration with agent-based interaction will be crucial. Researchers such as Volksen et al.[36] at Siemens have developed Cooperation-Ware as a framework for human-agent collaboration.

3. Applications of multi-agent systems in procurement and contracting/acquisition

In surveying the literature, there have only been a few multi-agent systems developed directly for the procurement and contracting/acquisition area. Mehra and Nissen[22] have designed an intelligent multi-agent supply chain management system using Gensym's Agent Development Environment, and Chen et al.[4] have built a negotiation-based multi-agent system for supply chain management. Steinmetz et al.[32] have designed an efficient anytime algorithm for multiple-component bid selection in automated contracting. In the logistics area, Satapathy et al.[31] have developed Distributed Intelligent Architecture for Logistics (DIAL). This is a multi-agent system designed to aid in real world logistics planning.

Business process management is an allied area relating to the acquisition management field. ADEPT[11] views a business process as a community of negotiating, service-provided agents. O'Brien and Wiegand[27] have developed an agent-based process management system architecture for workflow management. Additional work has been performed by Nissen[23,24,25,26] via an intelligent redesign agent called KOPeR.

Electronic commerce is a rapidly growing area, related to procurement and contracting, where multi-agent systems are being applied. Lee and Lee[17] have developed an intelligent agent-based competitive contract process using UNIK-AGENT. Zlotkin and Rosenschein[38] have worked on mechanisms for automated negotiation in state oriented domains. Tsvetovatyy and Gini[35] have developed MAGMA, a free-market agent architecture via automated purchasing and agent cooperation. The application of multi-agents for electronic commerce is a fertile growth area.

Other selected examples of multi-agent systems (non-acquisition related) that have been developed include Intelligent Agent Decision Support System (IADSS),[37] Autonomous Agents for Rock Island Arsenal (AARIA),[28] Remote Agent Experiment for Spacecraft Autonomy,[2] Internet-based multi-agent system for military training,[13] and Agent Inception System for visual modeling for agent-based applications.[7]

4. Multi-agent architecture for the pre-award phase of a contract

The CESA provides the primary source from which the multi-agent system's knowledge base has been developed.[18] CESA is a rule-based expert system developed at the US Naval Research Laboratory to help COTRs respond to questions relating to the pre-award phase of contract acquisition. MACS includes 119 rules of CESA's knowledge base covering the following areas:

- Adequacy of the PR package
    - What forms are needed in a PR package
        - Major Procurement
        - Supply
    - Justification and Approval (Sole Source)
        - What should be included in a sole source justification
        - What needs to be evaluated
        - Whether an Acquisition Plan is applicable
    - Evaluation
        - Evaluation weights and scoring
        - Evaluation criteria
        - Evaluation procedures
    - Synopsis
        - How to format the synopsis
        - Synopsis requirements for an 8a or Broad Agency Announcement response
        - Synopsis requirements for unsolicited proposals in R&D
    - Types of contracts
        - Firm fixed price
        - Cost plus fixed fee (CPFF)
        - Completion-type CPFF for hardware/software project; level of effort CPFF for services or on-going software development
        - Normally level of effort CPFF
        - Cost reimbursement/grant/student services contract

The web-based, multi-agent architecture presented in this paper for helping COTRs in the pre-award phase of a contract uses a six-agent architecture — a User Agent and five specialty agents that are entrusted with managing the various functions of CESA described above. The six agents represent a modified, brokered agency architecture. We say *modified*, brokered architecture because a User Agent functions as both an interface and a broker agent. That is, the User Agent interacts with the user/COTR to welcome the user, ask what pre-award questions the user has, and serves as the interface between the user/COTR and the other agents in the system. It will also (in future work) be coded with meta-knowledge about other agents in the system so that it can route user queries to specific agents for response. Thus, the typical three-tiered brokered architecture is reduced to two tiers.

The five specialty agents in the system each possess domain expertise about particular aspects of the pre-award phase. The specialty agents are dictated by the CESA knowledge base. The name of each specialty agent indicates its domain expertise and maps to the areas of the CESA rule base previously summarized as follows:

1. *Forms Agent.* This agent identifies the forms needed to complete the contract request based on characteristics of the contract.
2. *Justification Agent.* This agent indicates situations where a Justification and Approval is required to complete the PR.
3. *Evaluation Agent.* This agent provides information about evaluation weights, criteria, and procedures related to proposals.
4. *Synopsis Agent.* The agent is responsible for identifying situations where a contract synopsis is required for completion of the PR package.
5. *Types of Contracts Agent.* This agent identifies the nature of a contract based on contract conditions such as the source of contract and the nature of the work.

The specialty agents are self-contained (i.e. their knowledge bases are independent of the other specialty agents), and thus interaction between these specialty agents is not required. The brokered User Agent requires two-way feedback between itself and the specialty agents. It also has two-way feedback between itself and the user (ARO/COTR) so that responses can be forwarded and displayed to the user by the User Agent. As mentioned in Section 1, the User Agent is a Java servlet and the specialty agents are AgentBuilder® agents.

Agent communication and interaction proceeds in the following manner:

1. User Agent welcomes the User.
2. User sends a user request to the User Agent (currently via predetermined keywords selected from a list).
3. The User Agent determines if it understands the request and if so, then broadcasts the request to the Specialty Agents.
4. If the User Agent needs further clarification from the user, it then sends the request for further clarification back to the user.
5. The User then sends the "clarified" request to the User Agent who in turn sends it to the Specialty Agents.
6. If a Specialty Agent can answer the request, it sends the answer back to the User Agent, who in turn forwards it to the user.
7. If a Specialty Agent cannot answer the request, it sends the request back to the User Agent who then (if appropriate) forwards it to the user for further clarification.
8. If, after several rounds of clarification, none of the Specialty Agents can determine an answer to the request, they send this information to the User Agent who in turn sends this reply to the user.

Fig. 1 illustrates the system architecture and communication. Each specialty agent consists of four components, as shown in Fig. 2: *Perceptor/Effector*, *ACL communicator*, *Reasoner*, and *Modeler*. The *Perceptor/effector*is designed to communicate with the external world. Any data, other than ACL messages, is received and sent through this component. The *ACL communicator* is used to send and receive messages with other agents using an Agent Communication Language (KQML in this case). Incoming ACL messages are parsed and passed to the *Reasoner*. The *Reasoner*reasons with a message received from either *Perceptor* or *ACL communicator* to determine if any actions need to be performed to respond to the message. The *Modeler* is designed to store the domain knowledge of an agent, and MACS uses rules and frames to represent the domain knowledge of each specialty agent. Rules are used to represent retrieving strategies, and frames describe their information sources (forms, justification and approval statements, evaluation criteria, synopses, and contracts). This structure allows knowledge in the agents to be easily updated. For instance, whenever new forms or justification statements are released, new frames can easily be added to the Forms and Justification Agents.

Each agent has explicit goals. Its Modeler is responsible for guiding how to achieve the goals under varying circumstances. The specialty agents respond to incoming queries by presenting necessary information and/or requirements for ARO/COTRs. For example, the Evaluation Agent can assist a COTR with information regarding how to evaluate a project and what criteria or weights to use for evaluation of a contract. If a COTR has a question regarding "determining weights on evaluation criteria," the Evaluation Agent will reply with "You can develop your own weights on technical, qualifications, and cost criteria. Generally speaking, a weight of 40 percent (out of 100%) is given to cost." The COTR can input a variety of keywords pertaining to evaluation weights and criteria to which the Evaluation Agent will respond.

4.1. System interface

The user interface for this system is intended to support simple communications between the user and the system. In particular, the user will be expected to be aware of the characteristics of the contract under consideration. These characteristics are entered through a series of pull-down menus. Selections from these menus are then transported to the specialty agents in Agentbuilder as a string of keywords in a KQML message. The response from these agents is then sent back as a series of strings to the User Agent, and these are represented as recommendations from the multi-agent system.

Fig. 3 shows the input screen for the user agent and Fig. 4is the output screen. These figures are depicting a particular example that will be further discussed in Section 4.2. The interface can be described as follows:

1. The summary window in the input screen makes queries made by users available to them for easy recollection and revision. Each time the user makes a selection, the results

will be displayed using AND and OR conditions. The selection of AND and OR conditions is described below.

   (a) For the pull-down menus on the input screen labeled "Type of Contract" and "Contract Amount," the user may only select one keyword because choices in these pull-down menus are mutually exclusive.
   (b) For the remaining menus, the user may select multiple options and these will appear as AND conditions in the summary window.

2. The user may be allowed to deselect options from the summary window.
3. Selections in the summary window are sent as a string to the User Agent.
4. Responses from the specialty agents to the user agent are then displayed on the output screen. The summary window with the user's selections is also displayed on this screen. Once a response is sent, it is categorized according to the specific specialty agent from which the response was sent. Therefore, if multiple agents respond to the query, then their responses are appropriately categorized.
5. The output screen also allows the user to return to previous selections, start a new session, and exit from the interface.
6. When an appropriate response to a user's query is not found, the output screen displays the message "Sorry, but this agent does not have a response to your query."

4.2. The computerized multi-agent system

The agent architecture depicted in Fig. 1 has been computerized into a working multi-agent system. The user agent appears as in Fig. 3 and Fig. 4. The following figures are screen shots of the specialty agents in AgentBuilder. Specifically, Fig. 5 provides an overview of the specialty agents Synopsis, Contracts, Evaluation, Forms and Justification on the left-hand side of the screen. The test_agent is used to validate each specialty agent prior to linking it with the User Agent.

Fig. 6 and Fig. 7 depict different aspects of a sample rule in the Evaluation Agent of the multi-agent system. Fig. 6 shows the coding for rule 57 on the right-hand side of the screen. Under "WHEN" the conditions for firing the rule are listed-and these are the conditions given in Fig. 3. "THEN" provides the text displayed to the user after the "WHEN" conditions have been met. The test_agent is then shown in Fig. 7 with the KQML message that must be sent to the Evaluation Agent in order for rule 57 to fire. Once fired, the text from rule 57 is displayed to the user as previously shown in Fig. 4.

5. Summary

In this study, we have demonstrated the development of a multi-agent system that supports functions in defense acquisition tasks. Specifically, the goal of this multi-agent system is to help the COTR more easily and effectively answer questions relating to the pre-award phase of a contract over the Web. This is particularly useful because of the complex nature of the pre-award phase of contracting. Dividing the knowledge base into five areas of domain expertise via the specialty agents can enhance performance of the system by increasing the speed with which responses can be obtained from the specialty agents. Future testing and validation of the meta-knowledge encoded in the User Agent will have to be undertaken to further support this statement since responding specialty agents will be dependent on where the User Agent sends the COTR queries. In this regard, this system is one of the earliest ones in the field of defense contracting.

The study provides a foundation for enhancing this system such that it could be applied to domains other than contract acquisition. We envision that future work on this system in terms of the learning and natural language capabilities will support this generalization of our work. Furthermore, the integration of the system with the more dynamic DAD site will ensure the dynamic nature of this system and will reduce the risks of static systems that are often associated with traditional rule-based systems.

Future development of the system will include the integration of additional knowledge from the Ask a Professor questions (which are archived on the web) via the Defense Acquisition Deskbook. The emphasis of future work will be on the User Agent. First, the broadcast method for sending messages to the specialty agents will be converted to a routing system with meta-knowledge about each specialty agent's domain knowledge designed into the User Agent. This will be accomplished by parsing values from this input strings entered by users to direct queries to the various specialty agents. Second, additional functionality will be built into the User Agent. This increased functionality will involve three things as follows:

(a) *Distance Mechanism.* In preliminary efforts to provide the user with the best response, some distance mechanism may be built to identify the most suitable response to a user's query. This will be particularly useful in situations where there is no match between the user's keyword identification and Agentbuilder rules.

(b) *Learning.* The User Agent may be able to make inferences about a user's preferences based on similar interactions in the past. The user agent may also learn the nature of the query and direct it to the most appropriate agent to reduce redundancy in the queries.

(c) *Natural Language Abilities.* Eventually the user may enter a query in a window and some natural language functions will parse this to obtain a potential list of keyword that may be of interest to the user.

Acknowledgements

This work was partially funded by EARP'99 (External Acquisition Research Program) and a Fulbright Fellowship.

References

[1] K. Arisha, F. Ozcan, R. Ross, V. Subrahmanian, T. Eiter, S. Kraus. Impact: a platform for collaborating agents. IEEE Intelligent Systems, IEEE Computer Society Press, Los Alamitos, CA (1999)

[2] D. Bernard, G. Dorais, C. Fry, et al., Design of the Remote Agent Experiment for Spacecraft Autonomy, NASA Jet Propulsion Laboratory, Pasadena, CA, 1999.

[3] Bradshaw, J.R. Carpenter, R. Cranfill, R. Jeffers, L. Poblete, T. Robinson, A. Sun, Y. Gawdiak, I. Bichindaritz, K. Sullivan, Roles for Agent Technology in Knowledge Management: Examples from Applications in Aerospace and Medicine, White Paper, Boeing Information and Support Services, Seattle, WA, 1998.

[4] Y. Chen, Y. Peng, T. Finin et al., A Negotiation-based Multi-Agent System for Supply Chain Management, Working Paper,

Department of Computer Science, University of Maryland-Baltimore County, Baltimore, MD, 1998.

[5] J. Collis, H. Nwana, D. Ndumu, L. Lee, Zeus: an Advanced Toolkit for Engineering Distributed Multi-Agent Systems, British Telecom Laboratories, Marlesham Heath, UK, www.labs.bt.com/projects/agents/, 1998.

[6] S. DeLoach, Multiagent systems engineering: a methodology and language for designing agent systems, Proceedings of Agent-Oriented Information Systems (AOIS 99), 1999.

[7] B. Falchuk, A. Karmouch. Visual modeling for agent-based applications. IEEE Computer, IEEE Computer Society Press, Silver Spring, MD (1998)

[8] J. Ferber. Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison-Wesley, Harlow (1999)

[9] R. Flores-Mendez. Towards a standardized multi-agent system framework. ACM Crossroads, Association for Computing Machinery, New York (1999)

[10] P.H.J. Hendriks, D.J. Vriens. Knowledge-based systems and knowledge management: friends or foes? Information and Management Journal, 35 (1999)

[11] N. Jennings, P. Faratin, M. Johnson, T. Norman, P. O'Brien, M. Wiegand. Agent-based business process management. International Journal of Cooperative Information Systems (1996), p. 5

[12] A. Joshi, M. Singh. Multiagents systems on the Net. Special Issue of Communications of the ACM, Association for Computing Machinery, New York (1999)

[13] N. Joshi, V. Ramesh, Intelligent Agents for Internet-Based Military Training, www.stsc.hill.af.mil/crosstalk/, March 1998.

[14] Y. Labrou, T. Finin, Y. Peng. Agent communication languages: the current landscape. IEEE Intelligent Systems, IEEE Computer Society Press, Los Alamitos, CA (1999)

[15] S.E. Lander. Issues in multiagent design systems. IEEE Expert, 12 (2) (1997)

[16] C. Landauer, K. Bellman, Agent-Based Information Infrastructure, The Aerospace Corporation, Los Angeles, CA, April 5 1999.

[17] J.K. Lee, W. Lee. An intelligent agent-based competitive contract process: UNIK-AGENT. International Journal of Intelligent

Systems in Accounting, Finance, and Management, 7 (June) (1998)
[18] J. Liebowitz, CESA: A Historical Case Study and Lessons Learned, Review Quarterly, Defense Systems Management College, Virginia, Spring, 2000.
[19] J. Liebowitz (Ed.), The Knowledge Management Handbook, CRC Press, Boca Raton, FL (1999)
[20] J. Liebowitz. Building Organizational Intelligence: A Knowledge Management Primer, CRC Press, Boca Raton, FL (2000)
[21] J. Liebowitz, T. Beckman. Knowledge Organizations: What Every Manager Should Know, CRC Press, Boca Raton, FL (1998)
[22] A. Mehra, M. Nissen, Intelligent software supply chain agents using ADE, AAAI 98 Workshop on Software Tools for Developing Agents, American Association for Artificial Intelligence, www.gensym.com/successstories/aitools.html, 1998.
[23] M. Nissen, An intelligent agent for web-based process redesign, AAAI-99 Workshop, Orlando, FL, July 1999.
[24] M. Nissen, An Agent Federation for Supply Chain Integration, Naval Postgraduate School, Monterey, CA, 1999.
[25] M. Nissen, SPS and beyond: innovating acquisition through intelligent electronic contracting, Proceedings of the 1999 Acquisition Research Symposium, 1999.
[26] M. Nissen, The standard procurement system and beyond, Innovative Contracting: Practical Approaches, 1999.
[27] P. O'Brien, M. Wiegand. Agent based process management: applying intelligent agents to workflow. The Knowledge Engineering Review, 13 (June) (1998)
[28] H.V.D. Parunak, Practical and Industrial Applications of Agent-Based Systems, Industrial Technology Institute, 1998, www.iti.org.
[29] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. Tolone, A. Boughannam. A multi-agent system for enterprise integration. Journal of Applied Artificial Intelligence (2000) http://umbc.edu/~finin/papers/aai98.pdf
[30] D.W. Rasmus, Knowledge management: more than AI but less without it, PC AI, vol. 14, no. 2, Knowledge Technology Inc., Phoenix, AZ, March/April, 2000.
[31] G. Satapathy, S. Kumara, L. Moore, Distributed Intelligent Architecture for Logistics (DIAL), marie.iddr.ie.psu.edu, June 10 1996.

Knowledge-Based Systems, Vol. 13, No. 5 (October 2000): pg. 241-250. DOI.

[32] E. Steinmetz, J. Collins, M. Gini, B. Mobasher, An Efficient Anytime Algorithm for Multiple-Component Bid Selection in Automated Contracting, Working Paper, Department of Computer Science and Engineering, University of Minnesota, 1998.

[33] K. Sycara, Multiagent systems, AI Magazine, Association for Artificial Intelligence, Summer, 1998.

[34] K. Sycara, D. Zeng. Coordination of multiple intelligent software agents. International Journal of Cooperative Information Systems (1996), p. 00

[35] M. Tsvetovatyy, M. Gini, Toward a Virtual Marketplace: Architectures and Strategies, Department of Computer Science, Working Paper, University of Minnesota, Minneapolis, MN, 1998.

[36] G. Volksen, H. Haugeneder, A. Jarczyk, P. Loffler, Cooperation-Ware: Integration of Human Collaboration with Agent-Based Interaction, Siemens Corporation, Munich, Germany, 1996.

[37] H. Wang. Intelligent agent-assisted decision support systems: integration of knowledge discovery, knowledge analysis, and group decision support. Expert Systems With Applications Journal, 12 (1997)

[38] G. Zlotkin, J. Rosenschein. Mechanisms for automated negotiation in state oriented domains. Journal of Artificial Intelligence Research, 5 (1996)

Corresponding author

[a] Performed as a Fulbright Fellow, Center for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Canada.

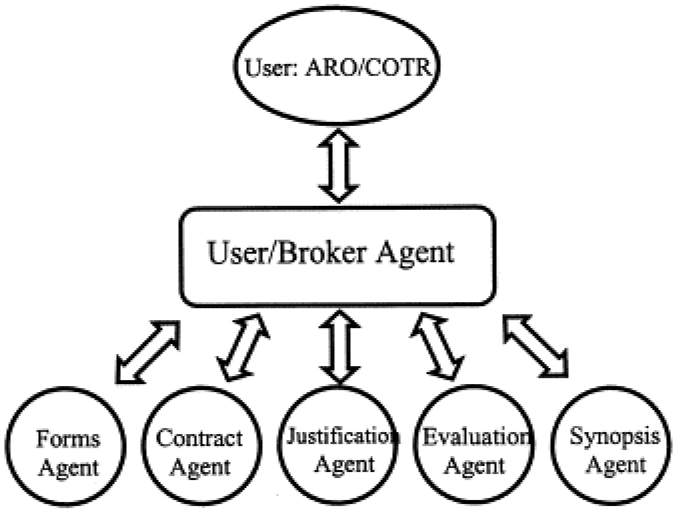

Fig. 1. Agent architecture and communication.

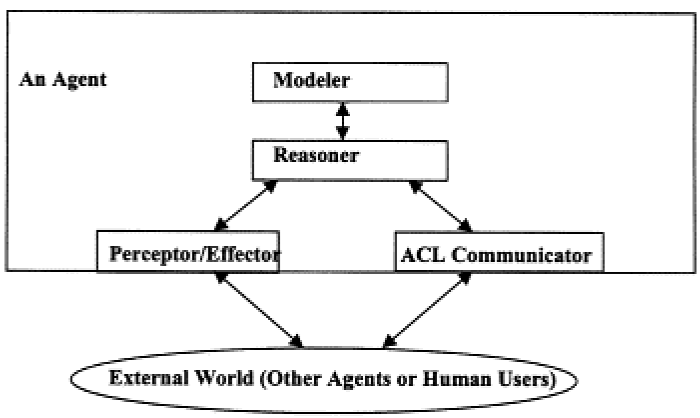

Fig. 2. Internal structure of an agent.

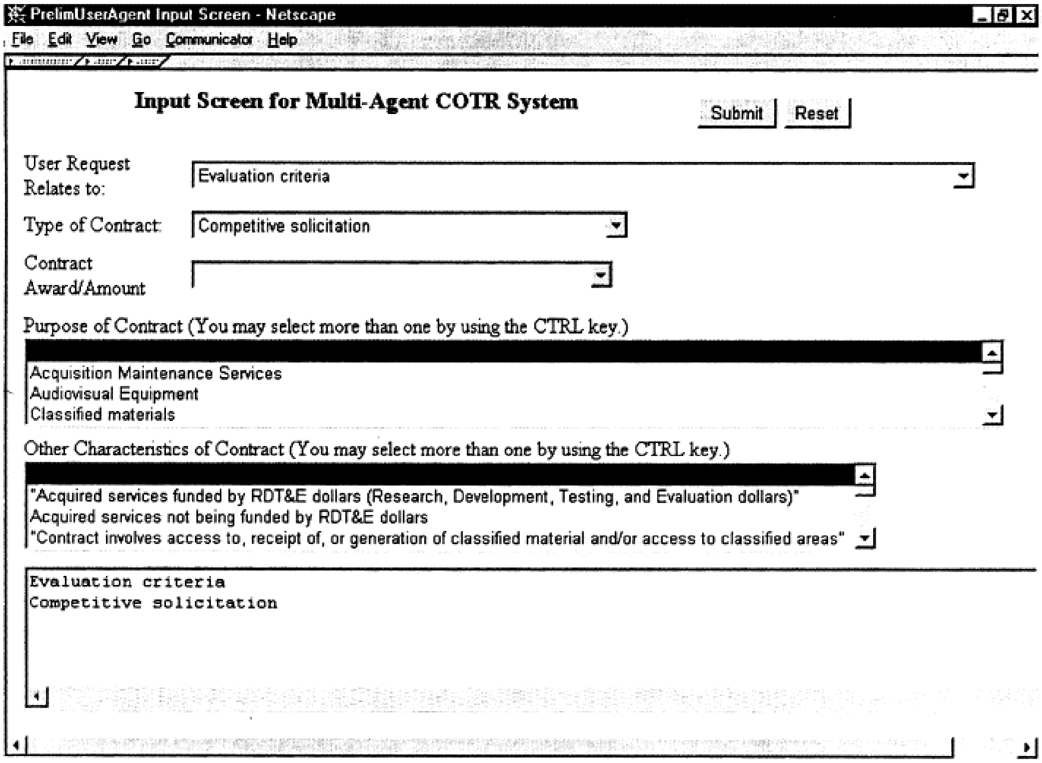

Fig. 3. User Agent input screen.

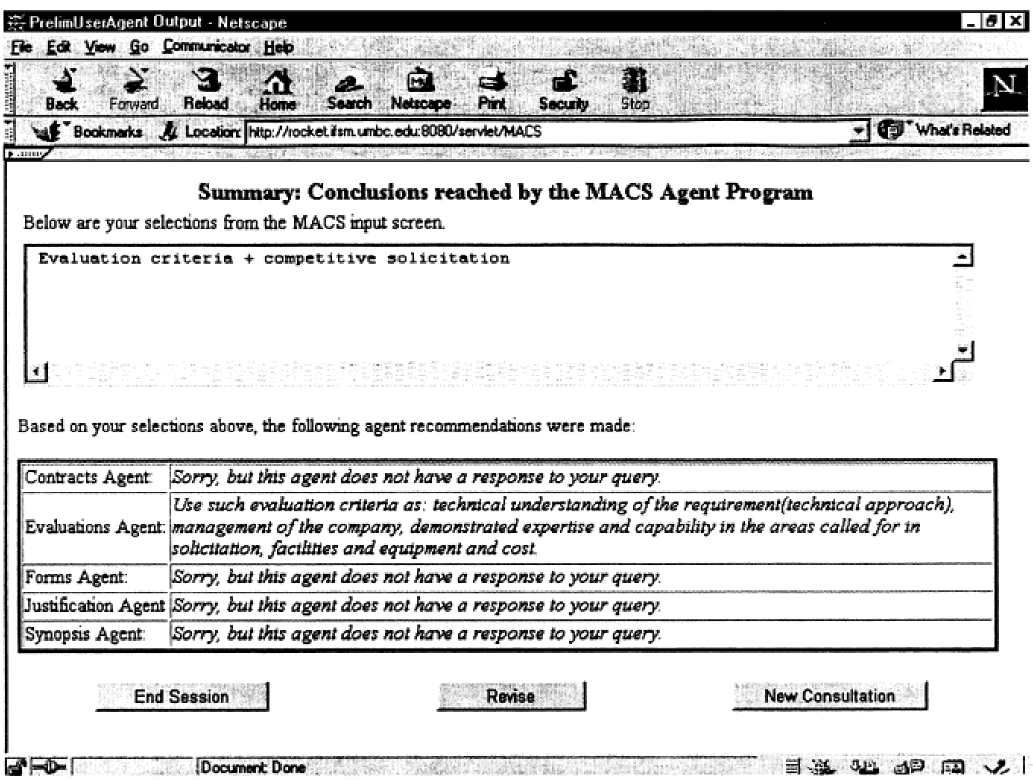

Fig. 4. User Agent output screen.

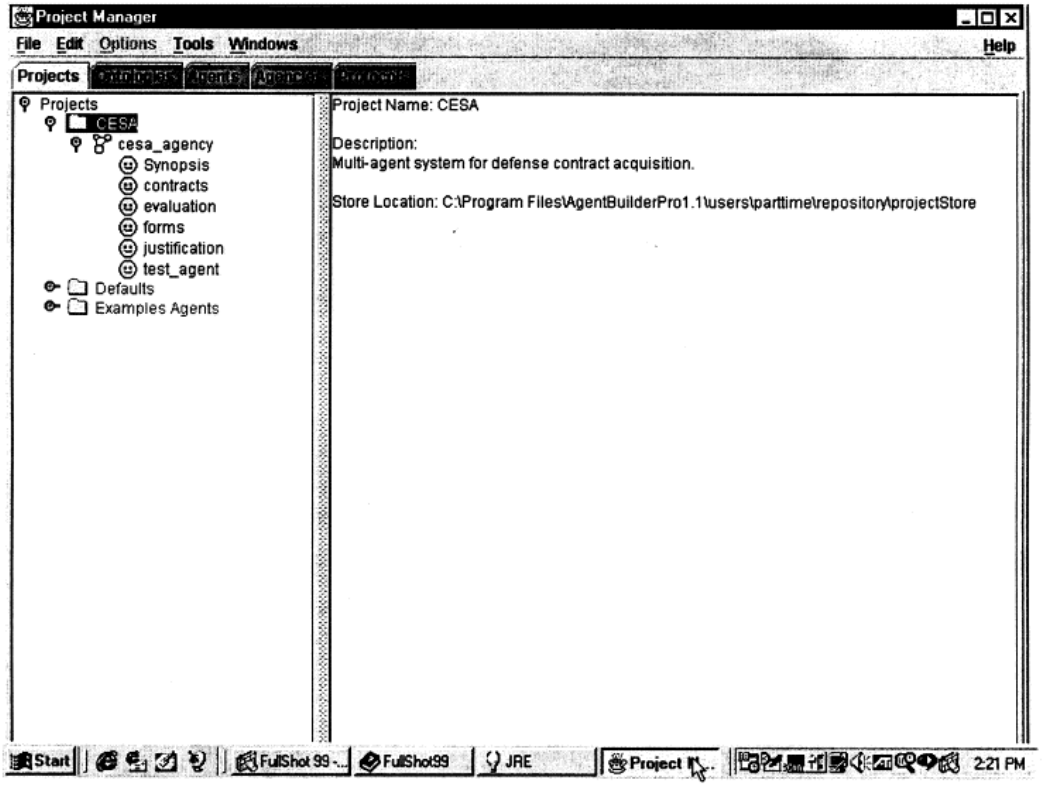

Fig. 5. Introductory screen displaying the CESA agency and agents.

Knowledge-Based Systems, Vol. 13, No. 5 (October 2000): pg. 241-250. DOI. This article is © Elsevier and permission has been granted for this version to appear in e-Publications@Marquette. Elsevier does not grant permission for this article to be further copied/distributed or hosted elsewhere without the express permission from Elsevier.

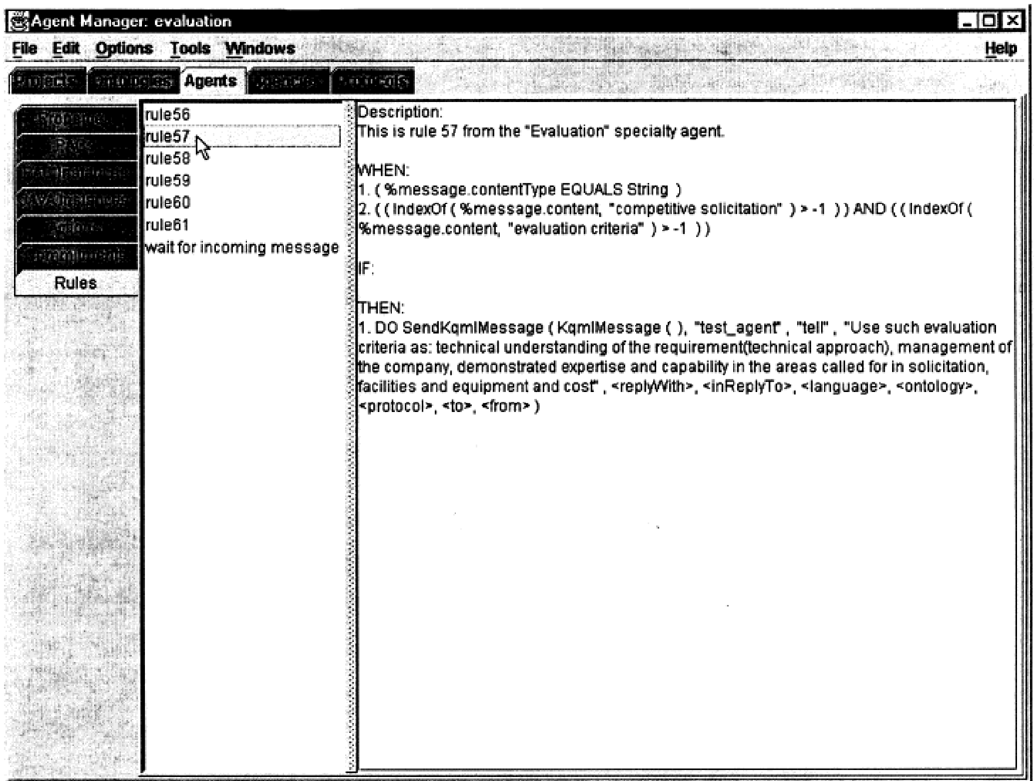

Fig. 6. Sample rule from the "Evaluation" specialty agent including inputs and outputs.

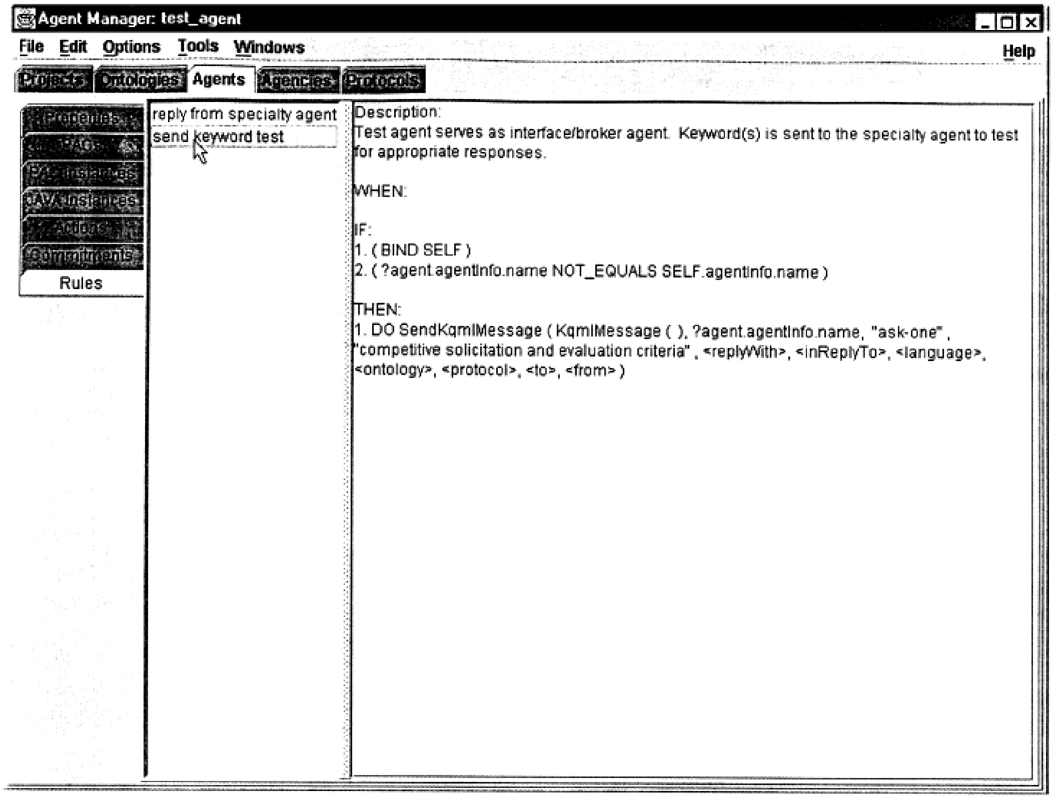

Fig. 7. Sample User Agent with input for "Evaluation" agent Rule 57 to fire.